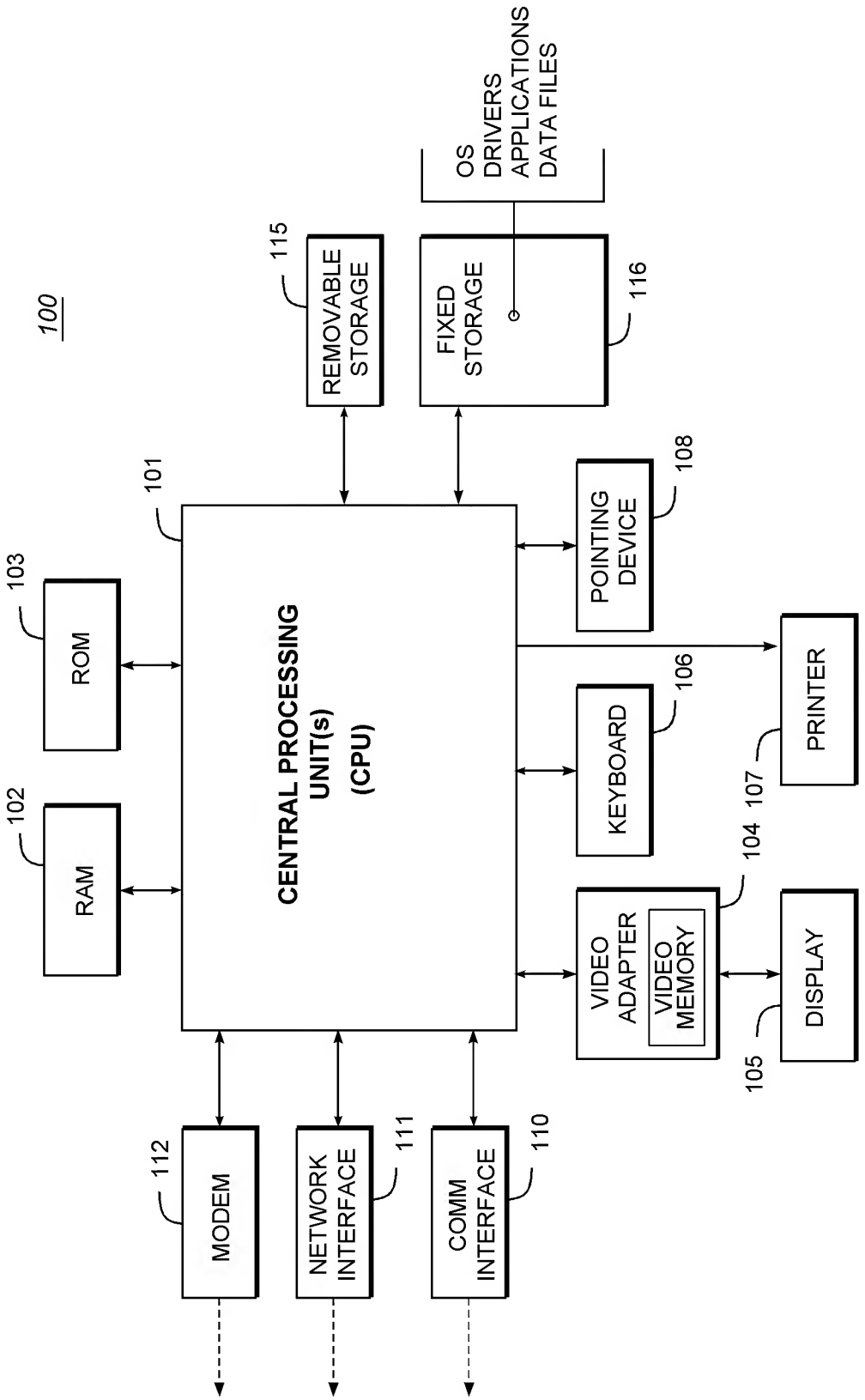


FIG. 1
(PRIOR ART)

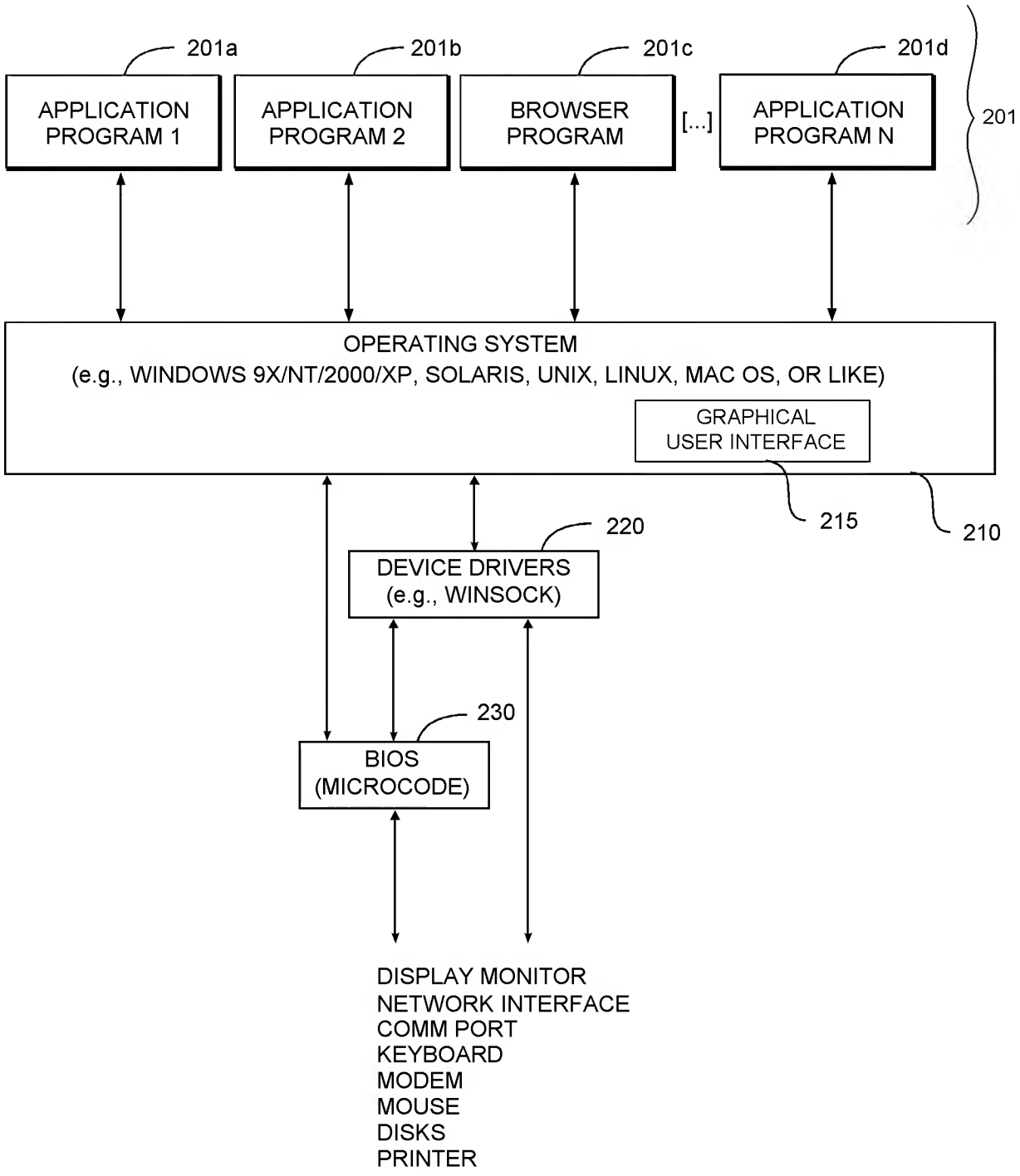

200
201a
201b
201c
201d
APPLICATION PROGRAM 1
APPLICATION PROGRAM 2
BROWSER PROGRAM
[...]
APPLICATION PROGRAM N
201
OPERATING SYSTEM
(e.g., WINDOWS 9X/NT/2000/XP, SOLARIS, UNIX, LINUX, MAC OS, OR LIKE)
GRAPHICAL USER INTERFACE
215
210
220
DEVICE DRIVERS
(e.g., WINSOCK)
230
BIOS
(MICROCODE)
DISPLAY MONITOR
NETWORK INTERFACE
COMM PORT
KEYBOARD
MODEM
MOUSE
DISKS
PRINTER

FIG. 2

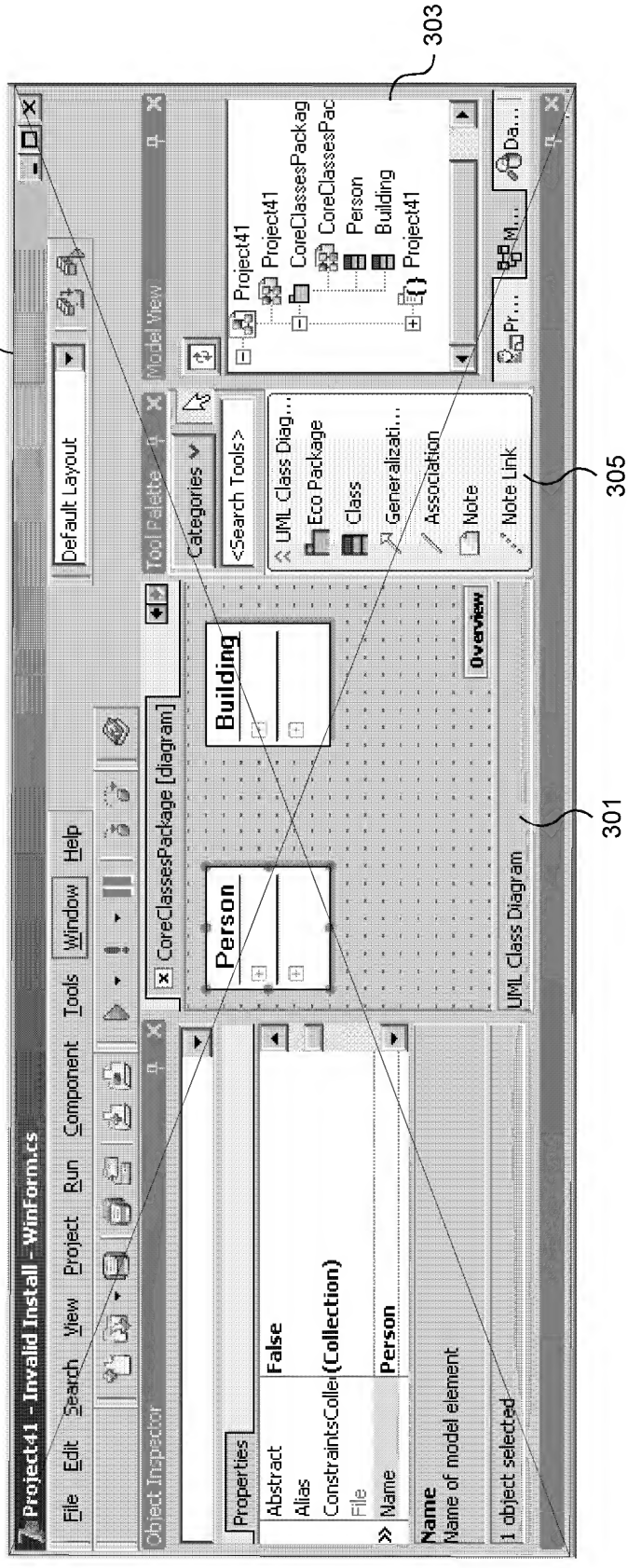
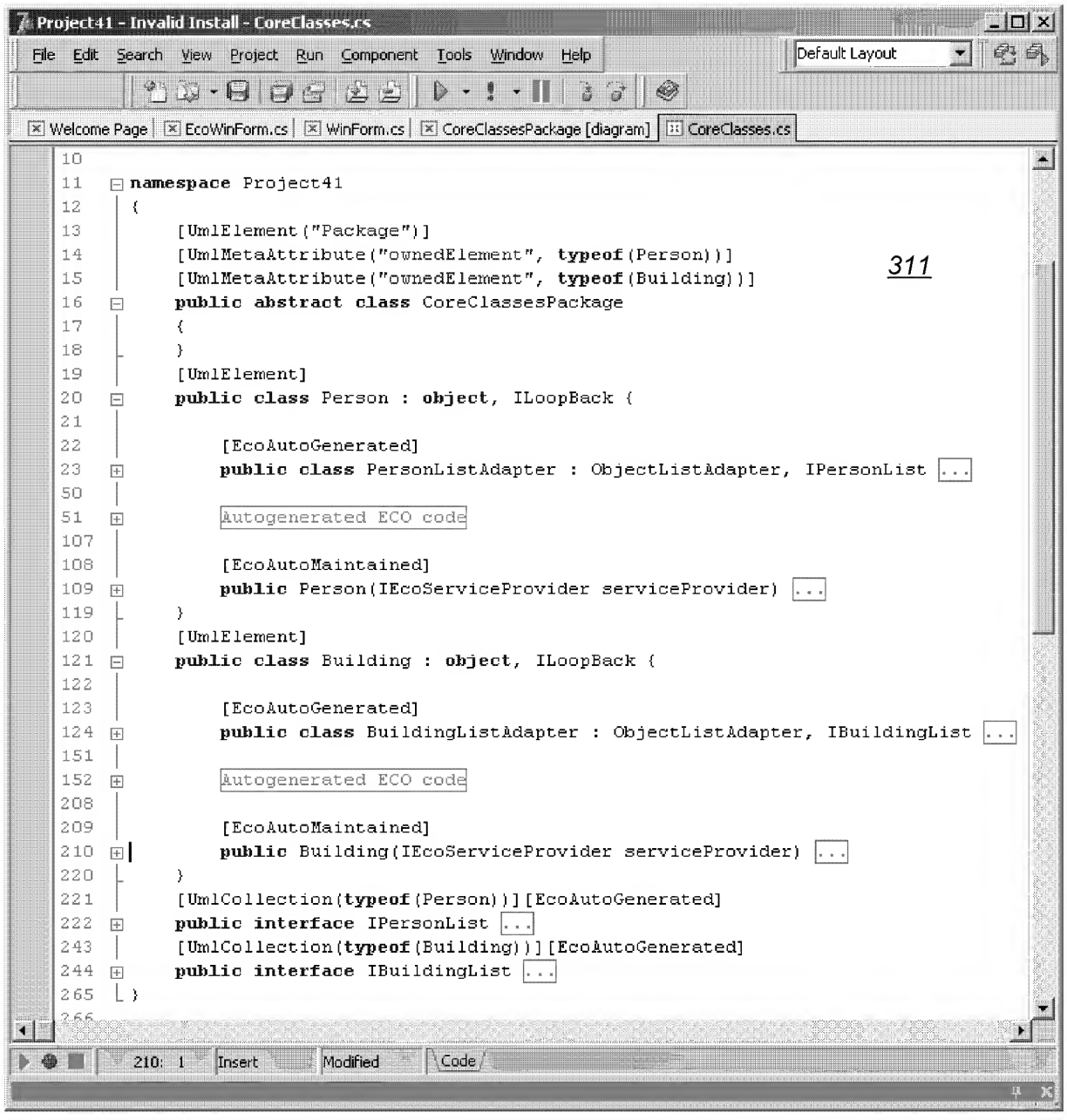

300
303
305
301
Project41 - Invalid Install - WinForm.cs
File Edit Search View Project Run Component Tools Window Help
Default Layout
Object Inspector
Properties
Abstract
Alias
ConstraintsColle (Collection)
File
Name
False
Person
Name
Name of model element
1 object selected
CoreClassesPackage [diagram]
Person
Building
Overview
UML Class Diagram
Tool Palette
Categories
<Search Tools>
UML Class Diag...
Eco Package
Class
Generalizati...
Association
Note
Note Link
Model View
Project41
Project41
CoreClassesPackag
CoreClassesPac
Person
Building
Project41

FIG. 3A




```
10
11  namespace Project41
12  {
13      [UmlElement("Package")]
14      [UmlMetaAttribute("ownedElement", typeof(Person))]
15      [UmlMetaAttribute("ownedElement", typeof(Building))]
16      public abstract class CoreClassesPackage
17      {
18      }
19      [UmlElement]
20      public class Person : object, ILoopBack {
21
22          [EcoAutoGenerated]
23          public class PersonListAdapter : ObjectListAdapter, IPersonList ...
50
51          Autogenerated ECO code
107
108         [EcoAutoMaintained]
109         public Person(IEcoServiceProvider serviceProvider) ...
119     }
120     [UmlElement]
121     public class Building : object, ILoopBack {
122
123         [EcoAutoGenerated]
124         public class BuildingListAdapter : ObjectListAdapter, IBuildingList ...
151
152         Autogenerated ECO code
208
209         [EcoAutoMaintained]
210         public Building(IEcoServiceProvider serviceProvider) ...
220     }
221     [UmlCollection(typeof(Person))][EcoAutoGenerated]
222     public interface IPersonList ...
243     [UmlCollection(typeof(Building))][EcoAutoGenerated]
244     public interface IBuildingList ...
265 }
```

FIG. 3B

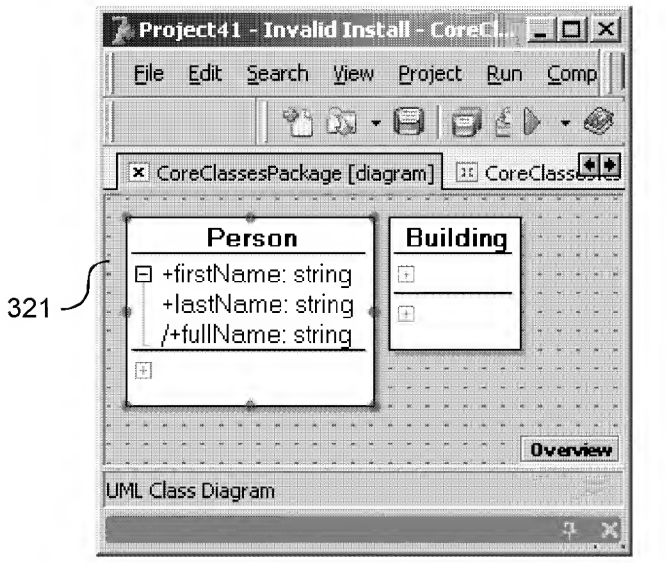

Project41 - Invalid Install - CoreCl...
File Edit Search View Project Run Comp
CoreClassesPackage [diagram]
CoreClasses
Person
+firstName: string
+lastName: string
/+fullName: string
Building
Overview
UML Class Diagram
321

FIG. 3C

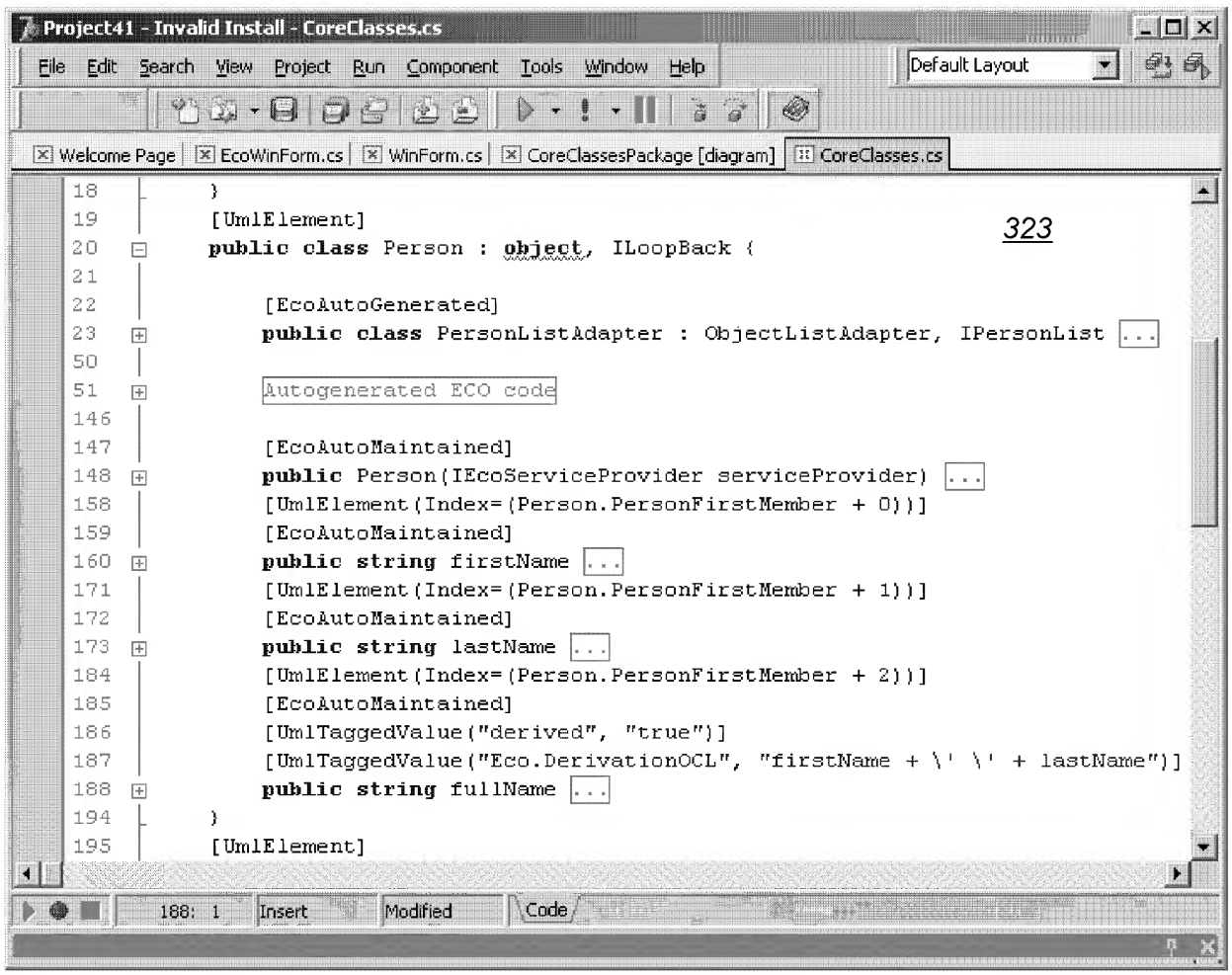

Project41 - Invalid Install - CoreClasses.cs
File Edit Search View Project Run Component Tools Window Help
Default Layout
Welcome Page
EcoWinForm.cs
WinForm.cs
CoreClassesPackage [diagram]
CoreClasses.cs
323
18 }
19 [UmlElement]
20 public class Person : object, ILoopBack {
21
22 [EcoAutoGenerated]
23 public class PersonListAdapter : ObjectListAdapter, IPersonList ...
50
51 Autogenerated ECO code
146
147 [EcoAutoMaintained]
148 public Person(IEcoServiceProvider serviceProvider) ...
158 [UmlElement(Index=(Person.PersonFirstMember + 0))]
159 [EcoAutoMaintained]
160 public string firstName ...
171 [UmlElement(Index=(Person.PersonFirstMember + 1))]
172 [EcoAutoMaintained]
173 public string lastName ...
184 [UmlElement(Index=(Person.PersonFirstMember + 2))]
185 [EcoAutoMaintained]
186 [UmlTaggedValue("derived", "true")]
187 [UmlTaggedValue("Eco.DerivationOCL", "firstName + \' \' + lastName")]
188 public string fullName ...
194 }
195 [UmlElement]
188: 1 Insert Modified Code

FIG. 3D

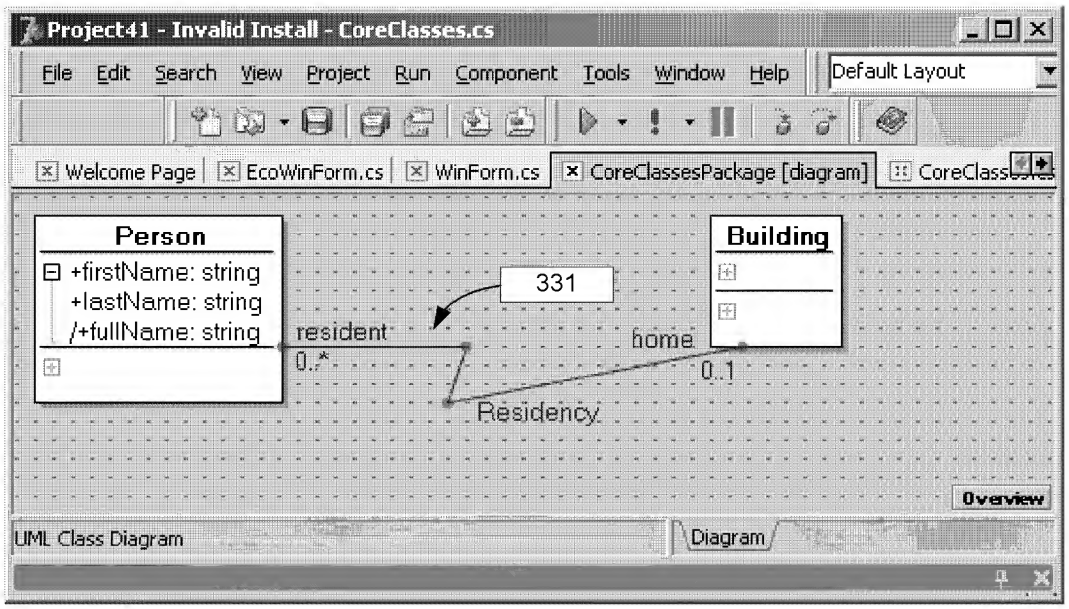
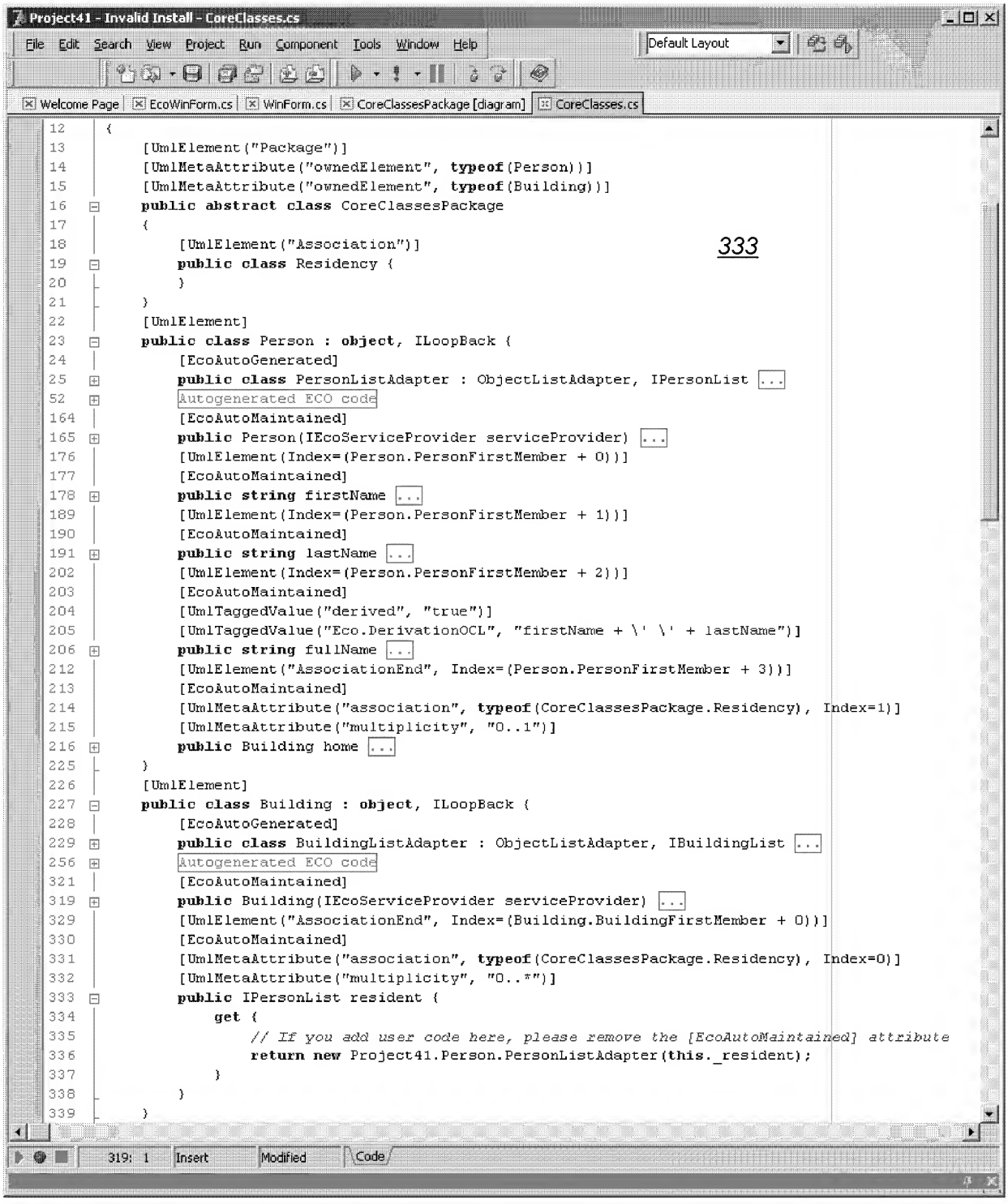

Project41 - Invalid Install - CoreClasses.cs
File Edit Search View Project Run Component Tools Window Help
Default Layout
Welcome Page
EcoWinForm.cs
WinForm.cs
CoreClassesPackage [diagram]
Person
+firstName: string
+lastName: string
/+fullName: string
Building
331
resident
0..*
home
0..1
Residency
Overview
UML Class Diagram
Diagram

FIG. 3E




```
12  {
13      [UmlElement("Package")]
14      [UmlMetaAttribute("ownedElement", typeof(Person))]
15      [UmlMetaAttribute("ownedElement", typeof(Building))]
16      public abstract class CoreClassesPackage
17      {
18          [UmlElement("Association")]
19          public class Residency {
20          }
21      }
22      [UmlElement]
23      public class Person : object, ILoopBack {
24          [EcoAutoGenerated]
25          public class PersonListAdapter : ObjectListAdapter, IPersonList ...
52          Autogenerated ECO code
164         [EcoAutoMaintained]
165         public Person(IEcoServiceProvider serviceProvider) ...
176         [UmlElement(Index=(Person.PersonFirstMember + 0))]
177         [EcoAutoMaintained]
178         public string firstName ...
189         [UmlElement(Index=(Person.PersonFirstMember + 1))]
190         [EcoAutoMaintained]
191         public string lastName ...
202         [UmlElement(Index=(Person.PersonFirstMember + 2))]
203         [EcoAutoMaintained]
204         [UmlTaggedValue("derived", "true")]
205         [UmlTaggedValue("Eco.DerivationOCL", "firstName + \' \' + lastName")]
206         public string fullName ...
212         [UmlElement("AssociationEnd", Index=(Person.PersonFirstMember + 3))]
213         [EcoAutoMaintained]
214         [UmlMetaAttribute("association", typeof(CoreClassesPackage.Residency), Index=1)]
215         [UmlMetaAttribute("multiplicity", "0..1")]
216         public Building home ...
225     }
226     [UmlElement]
227     public class Building : object, ILoopBack {
228         [EcoAutoGenerated]
229         public class BuildingListAdapter : ObjectListAdapter, IBuildingList ...
256         Autogenerated ECO code
321         [EcoAutoMaintained]
319         public Building(IEcoServiceProvider serviceProvider) ...
329         [UmlElement("AssociationEnd", Index=(Building.BuildingFirstMember + 0))]
330         [EcoAutoMaintained]
331         [UmlMetaAttribute("association", typeof(CoreClassesPackage.Residency), Index=0)]
332         [UmlMetaAttribute("multiplicity", "0..*")]
333         public IPersonList resident {
334             get {
335                 // If you add user code here, please remove the [EcoAutoMaintained] attribute
336                 return new Project41.Person.PersonListAdapter(this._resident);
337             }
338         }
339     }
```

FIG. 3F

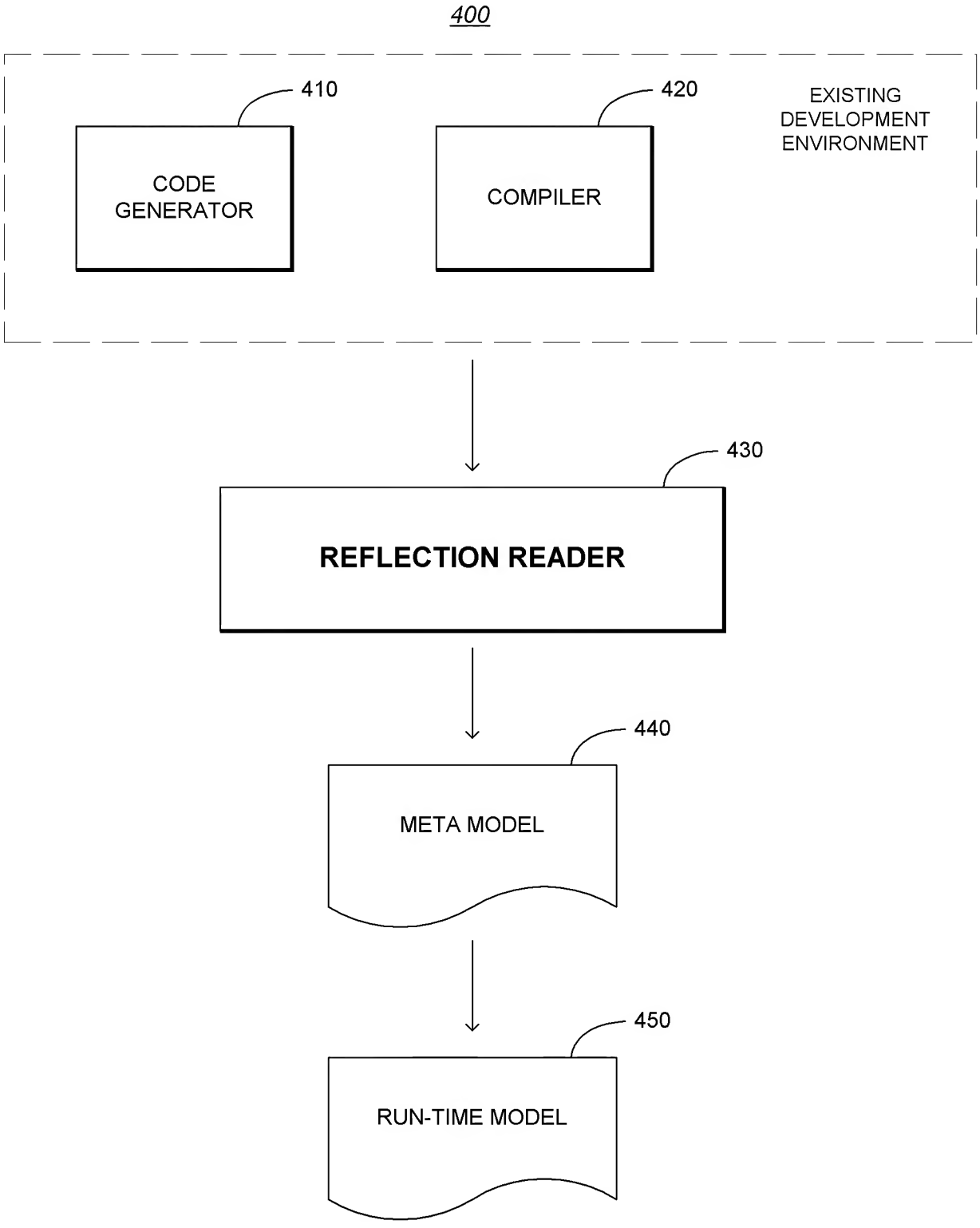

400
410
CODE GENERATOR
420
COMPILER
EXISTING DEVELOPMENT ENVIRONMENT
430
REFLECTION READER
440
META MODEL
450
RUN-TIME MODEL

FIG. 4

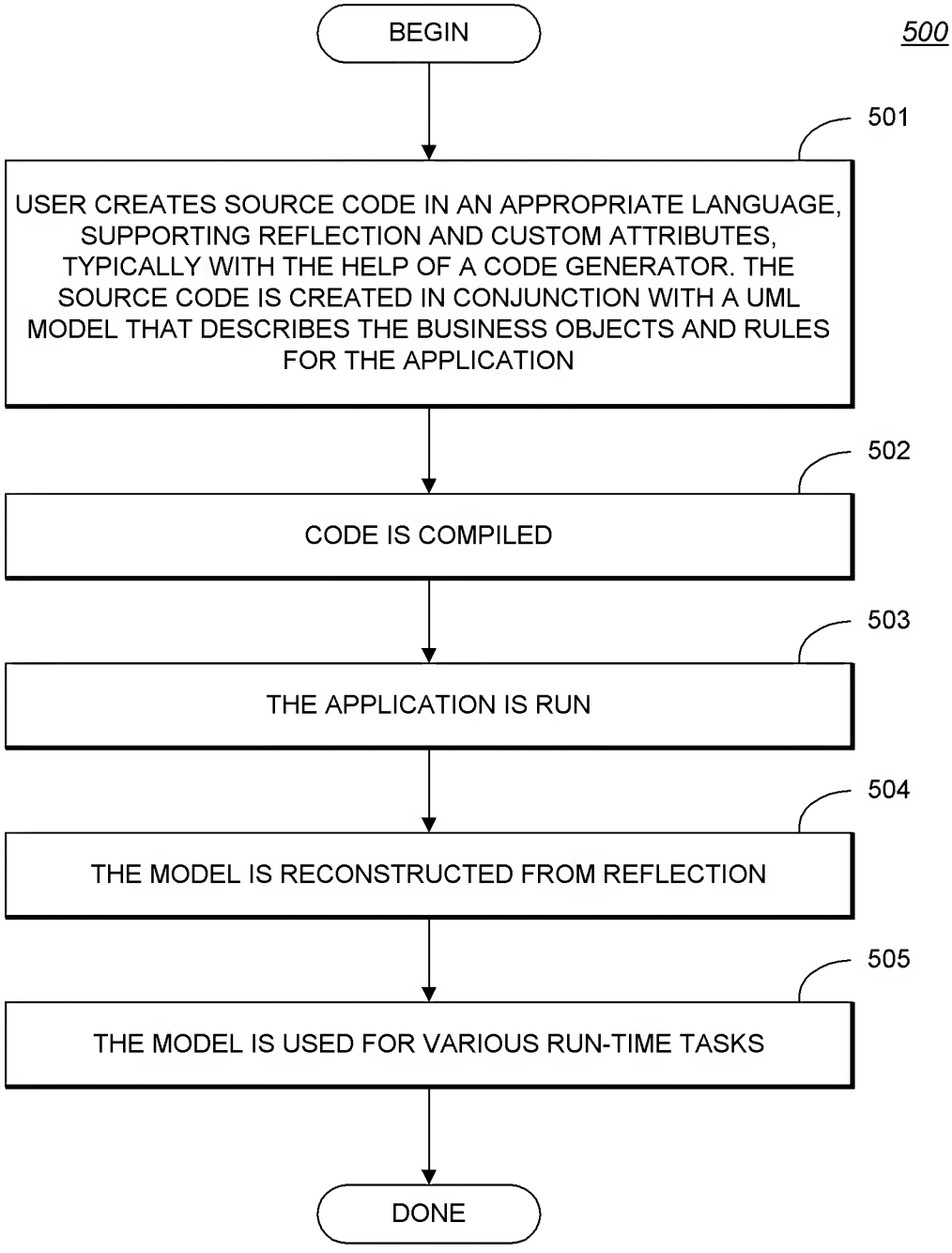

BEGIN
500
501
USER CREATES SOURCE CODE IN AN APPROPRIATE LANGUAGE, SUPPORTING REFLECTION AND CUSTOM ATTRIBUTES, TYPICALLY WITH THE HELP OF A CODE GENERATOR. THE SOURCE CODE IS CREATED IN CONJUNCTION WITH A UML MODEL THAT DESCRIBES THE BUSINESS OBJECTS AND RULES FOR THE APPLICATION
502
CODE IS COMPILED
503
THE APPLICATION IS RUN
504
THE MODEL IS RECONSTRUCTED FROM REFLECTION
505
THE MODEL IS USED FOR VARIOUS RUN-TIME TASKS
DONE

FIG. 5A

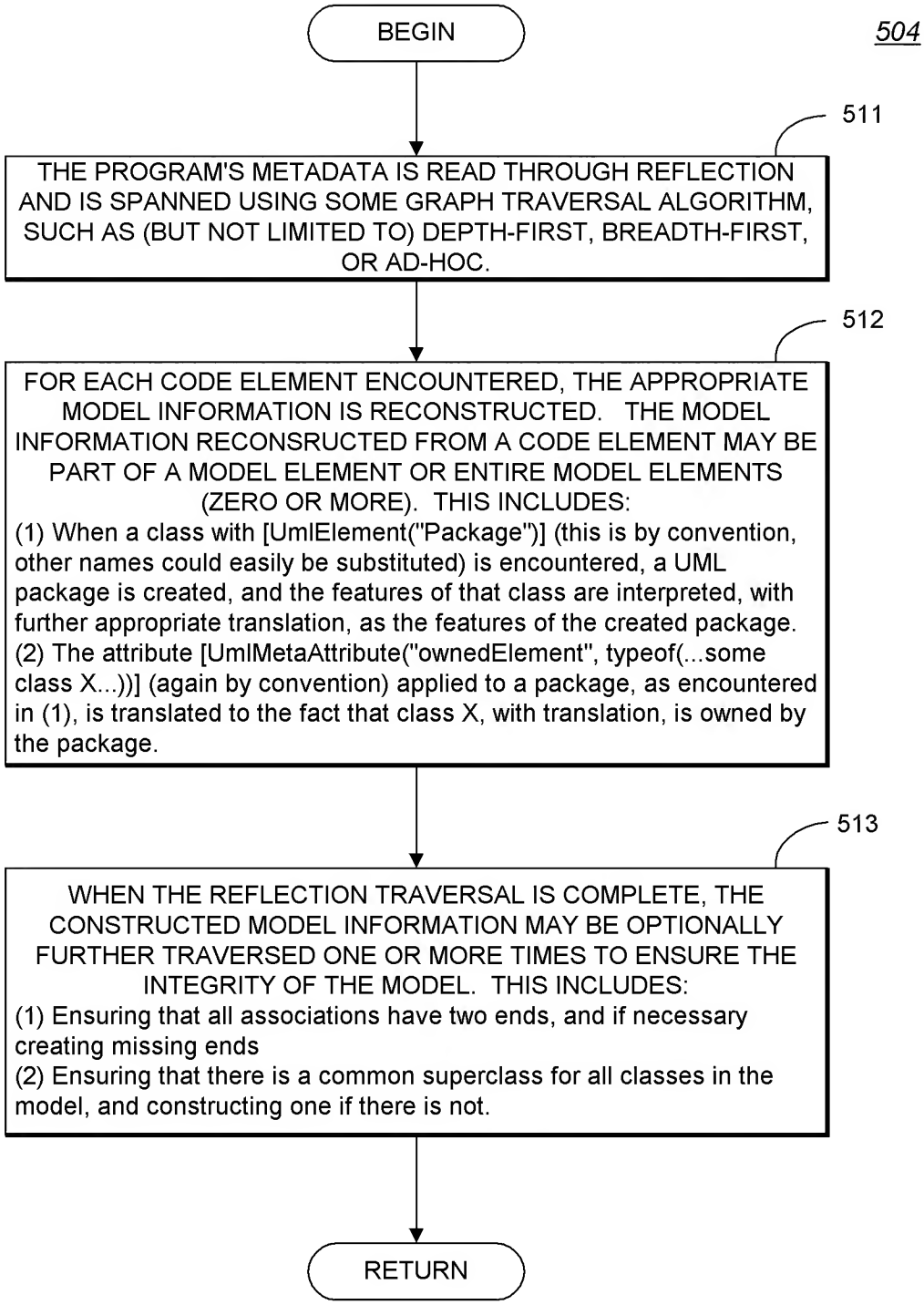

BEGIN
504
511
THE PROGRAM'S METADATA IS READ THROUGH REFLECTION AND IS SPANNED USING SOME GRAPH TRAVERSAL ALGORITHM, SUCH AS (BUT NOT LIMITED TO) DEPTH-FIRST, BREADTH-FIRST, OR AD-HOC.
512
FOR EACH CODE ELEMENT ENCOUNTERED, THE APPROPRIATE MODEL INFORMATION IS RECONSTRUCTED. THE MODEL INFORMATION RECONSRUCTED FROM A CODE ELEMENT MAY BE PART OF A MODEL ELEMENT OR ENTIRE MODEL ELEMENTS (ZERO OR MORE). THIS INCLUDES:
(1) When a class with [UmlElement("Package")] (this is by convention, other names could easily be substituted) is encountered, a UML package is created, and the features of that class are interpreted, with further appropriate translation, as the features of the created package.
(2) The attribute [UmlMetaAttribute("ownedElement", typeof(...some class X...))] (again by convention) applied to a package, as encountered in (1), is translated to the fact that class X, with translation, is owned by the package.
513
WHEN THE REFLECTION TRAVERSAL IS COMPLETE, THE CONSTRUCTED MODEL INFORMATION MAY BE OPTIONALLY FURTHER TRAVERSED ONE OR MORE TIMES TO ENSURE THE INTEGRITY OF THE MODEL. THIS INCLUDES:
(1) Ensuring that all associations have two ends, and if necessary creating missing ends
(2) Ensuring that there is a common superclass for all classes in the model, and constructing one if there is not.
RETURN

FIG. 5B

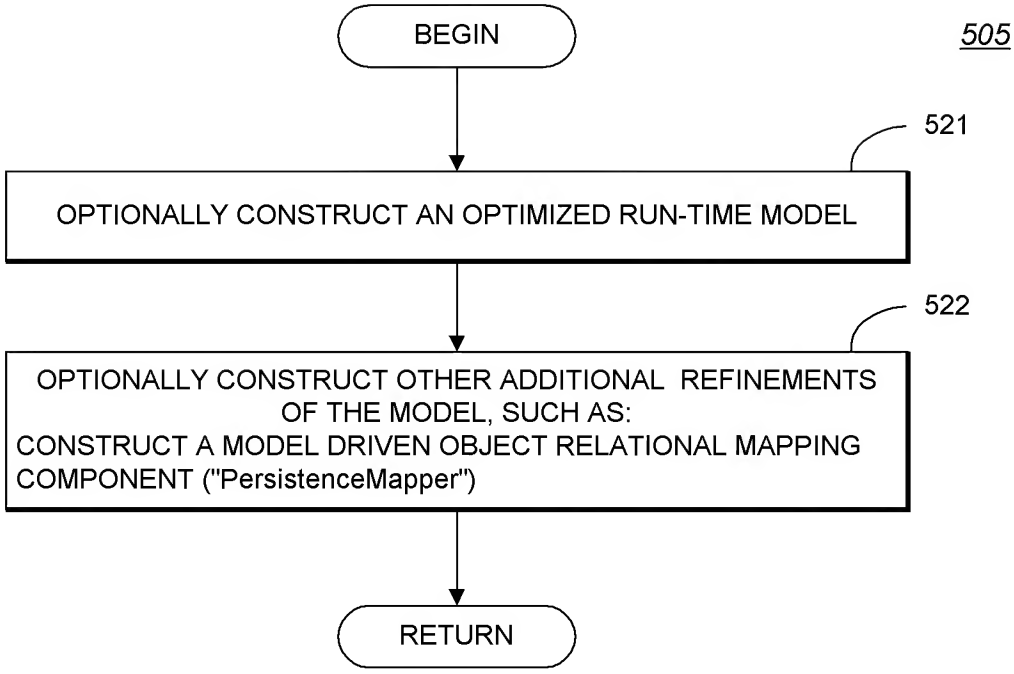

BEGIN
505
521
OPTIONALLY CONSTRUCT AN OPTIMIZED RUN-TIME MODEL
522
OPTIONALLY CONSTRUCT OTHER ADDITIONAL REFINEMENTS OF THE MODEL, SUCH AS:
CONSTRUCT A MODEL DRIVEN OBJECT RELATIONAL MAPPING COMPONENT ("PersistenceMapper")
RETURN

FIG. 5C